UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/072,323	10/23/2001	Jeffrey L. Kodosky	5150-46000	3183

35690 7590 08/22/2005

MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
P.O. BOX 398
AUSTIN, TX 78767-0398

EXAMINER	
BAYERL, RAYMOND J	

ART UNIT	PAPER NUMBER
2173	

DATE MAILED: 08/22/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/072,323	KODOSKY, JEFFREY L.	
	Examiner	Art Unit	
	Raymond J. Bayerl	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 June 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 October 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 6 – 9, 11 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

At line 5 of claim 6, "the graphical program" lacks clear antecedent basis. Only "a program" appears previously.

In claim 11, it is not clear how "interactively operating the user interface element without requiring execution of a program" can occur. Some form of "program", though perhaps not the one intended by applicant in the claim, **has** to be performed to exercise such control.

3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1 – 47 are rejected under 35 U.S.C. 102 (b) as being anticipated by Kodosky et al. ("Kodosky"; US #5,301,301).

As per claim 1 (method) and 19 (memory medium), Kodosky discloses a computer implemented method for associating a first block diagram with a user interface element that reads upon each and every feature claimed.

displaying the user interface element is taught by Kodosky, when a LabVIEW user constructs a virtual instrument from building blocks by defining an icon and connector for the virtual instrument. The connector associates each terminal of the

connector with an indicator or control on the front panel of the VI (see col. 4, lines 16-22).

receiving user input specifying the first block diagram to associate with the user interface element, wherein the first block diagram includes a plurality of nodes visually indicating functionality of the user interface element is taught by Kodosky's technique of creating a virtual instrument building block by defining an icon and connector for a virtual instrument. In associating each terminal of the connector with an indicator or control **on the front panel** of the VI (see col. 4, lines 16-22), and once the icon and the connector have been constructed, it is then possible to use the VI as a node in a diagram (see col. 4, lines 23-25 and see Figs. 2 for nodes icon 44, 48, and 50). For example, Fig. 3 is a screen display showing an exemplary data flow block diagram which can be produced by the VI system (see col. 2, lines 23-25). The block diagram employed in the programming environment of Kodosky serves as the basis upon which the front panel user interface objects will perform. In so doing, they are used in "indicating functionality of" that "user interface element".

associating the first block diagram with the user interface element, wherein the first block diagram is operable to control functionality of the user interface element is taught by Kodosky's use of a library of **function icons**, each representing a **mathematical operation to be performed on specified input data to generate output data** (see col. 8, lines 53-55). This means that the functions designated from the library are such that they control the way in which the front panel objects handle data, and the overall operation of the operator interface in Kodosky. In so doing, the

functions obtained in Kodosky's visual programming are instrumental, in their iconic arrangement in a block diagram, "to control functionality of the user interface element".

As per claim 2, a user interface control or a user interface indicator is taught by Kodosky's association of a created virtual instrument with an indicator or control on the front panel of the VI (see col. 4, lines 16-22).

As per claims 3 (method) and 20 (memory medium), the first block diagram which comprises a plurality of interconnected nodes that visually indicate functionality of the user interface element is taught by Kodosky, as seen in Figs. 3 and Fig. 7, which are screen displays showing an exemplary data flow block diagram and the TEMP SYS including a plurality of node icons. These control the graphical programming results from a "user interface" standpoint, and thus, the UI's "functionality".

As per claim 4's arranging the plurality of nodes on a display and interconnecting the plurality of nodes in response to the user input, Kodosky teaches that LabVIEW users **create** VIs which can be used as building blocks in other VIs. VIs are analogous to subroutines so it is useful to display the hierarchical relationship of VIs (see col. 4, lines 54-57; Fig. 7 shows an illustrative diagram). In creation of a block diagram, arranging and interconnecting both occur "in response to user input".

As per claim 5's graphical data flow diagram, Kodosky shows that LabVIEW extends the concept to the **data flow** programming environment (see col. 3, lines 38-41).

As per claims 6 (method) and 21 (memory medium), the limitations of including the user interface element in a program, so that upon execution, the first block diagram

is controlling functionality of the user interface element, Kodosky has a connector for user associate each terminal of the connector with an indicator output or control input on the front panel of the VI (see col. 4, lines 20-22). The user may choose to enable the acquisition and analysis processes independently (see col. 5, lines 17-25). Such incorporation of a block diagram means that it is made part of the overall Kodosky "graphical program", where the block diagram directs "user interface" "functionality".

As per claims 7 (method) and 22 (memory medium), displaying the user interface element and receiving user input to it, with the block diagram controlling functionality of the user interface element in response to the user input received, please note that Kodosky's LabVIEW user associates a block diagram with an indicator or control on the front panel of the VI (see col. 4, lines 16-22) and the user set high and low limits for the temperature, to activate warning messages when the temperature goes out of those limits. The histogram parameters may be adjusted. The user may choose to enable the acquisition and analysis processes independently and may set the update rate for the displays (see col. 5, lines 17-26). All of this means that the block diagram is fundamental in responding to user interface input.

As per claim 8's displaying the first block diagram, this occurs when the block diagram-editing user in Kodosky has a visual display of the icons that comprise the block diagram. Upon editing that first block diagram (claim 9), Kodosky's user achieves "changing how first block diagram controls functionality of the user interface element", as it appears at the front panel.

As per claims 10 (method) and 23 (memory medium), when receiving user input to the user interface element, the first block diagram is shown as responding to the user input in Kodosky's technique of the user set high and low limits for the temperature and which activate warning messages when the temperature goes out of those limits. The histogram parameters may be adjusted (see col. 5, lines 17-23) and the user may choose to enable the acquisition and analysis processes independently and may set the update rate for the displays (see col. 5, lines 23-26). All of this input at the front panel side of Kodosky is taken in, and the block diagram determines the resultant display "functionality", in such operations as these.

As per claim 11, interactively operating the user interface element without requiring execution of a program to operate the user interface element is taught by Kodosky, since in LabVIEW there are two ways to produce constants: 1) create a front panel control, set its value and then hide it so only the terminal appears on the block diagram; 2) place a control **directly on the diagram** rather than on the control panel (see col. 5, lines 36-40). By so doing, Kodosky teaches the direct implementation of controls, apart from "a program to operate the user interface element" that might otherwise be put in place.

As per claim 12's including the user interface element in a graphical program in response to user input, Kodosky teaches defining the connector for user associate each terminal of the connector with an indicator output or control input on the front panel of the VI (see col. 4, lines 20-22) and the user set high and low limits for the temperature and which activate warning messages when the temperature goes out of those limits.

This means that the front panel objects in Kodosky are included in a graphical program, whose execution means that the "first block diagram" controls "functionality".

As per claim 13's connection to a second block diagram, in handling the dataflow block diagrams (see col. 3 line 68 to col. 4 line 1), a LabVIEW user can create VIs which can be used as building blocks in other VIs (see col. 4, lines 54-55). The connection to other VI instances then requires connection to "a second block diagram". Furthermore, the interconnection means that "the first block diagram is accessible from the second block diagram" (claim 14) to which it is connected, in the building blocks arrangement.

Independent claim 15's "including a user interface element in a graphical program" (see also similar independent claim 25) occurs when Kodosky implements a front panel object that corresponds to a "first block diagram". Then, in using a VI as building blocks, it is possible that "a second block diagram" be connected to it, so that it "implements second functionality of the graphical program" via the influence the connected VI has on the "program" that is established in relation to the front panel, at the same time that "the first block diagram...is operable to control the first functionality of the user interface element". A similar statement may be made, regarding the connection of "a main block diagram" to "the block diagram associated with the user interface element", in independent claim 18. See further the similarity to Kodosky in the "main block diagram"/"block diagram associated with the user interface element", that implements two "functionality" settings in independent claim 24.

As per independent claim 16, a distinct possibility in Kodosky is that "a plurality of user interface elements" will be supported in a single VI implementation. This will be "a

plurality of primitive user interface controls" that are derived from LabVIEW's "application development environment", as in claim 17.

As per claims 26, 30, 32, 34, 38 the first block diagram in Kodosky is operable to change appearance of the user interface, since the resultant effect of the "block diagram" is to change the appearance of the calculated results of functions, etc. at the front panel. This also is operable to change a manner in which data is displayed in the user interface (claims 27, 31, 35), since the results affect this manner, "in response to the user input" (claims 33, 39) which will cause "the first block diagram" to "process the user input" (claims 42, 46) and have a resultant effect upon the "visual appearance" (claims 43, 47).

As per claim 28's copying the user interface from a first graphical program to a second graphical program and programmatically **including** the first block diagram in the second graphical program (see also claim 36), please note further Kodosky's technique of user access by **selecting the Graphical Array and Graph from Functional Palette** (see col. 8, lines 31-33 and see Fig. 19) **from users created VIs for building blocks in other VIs** (see col. 4, lines 41-42). **VIs are analogous to subroutines** so it is useful to display the hierarchical relationship of VIs (see col. 4, lines 55-57). In all of this user creation, it is evident that copying (as in subroutines) will occur, with the attached "block diagram" needing to be copied as well. This also results in "**associating** the first block diagram with the second graphical program", as in claims 29, 37, so that "the user interface element" moderated by "the first block diagram" "receives the data from the

second block diagram" as in claims 40, 44 to "control a visual appearance of the user interface element based on the data" (claims 41, 45).

5. Applicant's arguments filed 9 June 2005 have been fully considered but they are not persuasive.

In asserting, beginning at page 12 of the remarks, that "Kodosky does not teach associating a block diagram with a user interface element, wherein the block diagram includes a plurality of nodes visually indicating functionality of the user interface element and wherein the block diagram associated with the user interface element is operable to control functionality of the user interface element", applicant conducts a discussion and review of Kodosky that extends through page 14, where applicant states that while "the block diagram of a VI may visually indicate that data is received from a front panel...and may visually indicate that data is passed to a front panel indicator user interface element", "this is not at all the same as a block diagram visually indicating functionality of a user interface element or controlling functionality of a user interface element".

However, the underlying "block diagram" of Kodosky is essential in defining just how the "user interface" at the front panel will handle and respond to user input. Thus, the "user interface element" "functionality" is established, when the term "functionality" is given a reasonably broad interpretation. Kodosky indeed **will** affect "the manner in which the data is displayed" (see also the page 17 argument concerning claim 27), if for no other reason than the particular result that is produced will bespeak "functionality" and a "manner" of response and display. In producing particular results, an

"appearance of the user interface element" (argument concerning claim 41, page 18) is directly affected by the "block diagram"'s structure and resultant operations.

Concerning claim 15, applicant argues at page 16 that "Kodosky simply does not teach or suggest a user interface element that has an associated block diagram", as part of "a graphical program that has a main block diagram" such that "the block diagram...is separate from the main block diagram". However, in using the building blocks approach of Kodosky that is noted above, this form of interconnection is implemented between VI units. Larger block diagrams are constructed of smaller ones in Kodosky, this being a key feature of the LabVIEW environment, and "a second block diagram" can thereby influence a "first block diagram", to answer applicant's page 17 arguments concerning claims 13, 15. This can be accomplished by "copying", to address the argument concerning claim 28 at page 18.

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A further search indicated the relevancy of King et al. (US #2003/0071845 A1) and Kodosky et al. (US #2002/0083413 A1) both of which relate a block diagram to a user interface element.

7. In responding to this office action, please note that the examiner of record for the above identified application has changed. Any further inquiry concerning this communication or earlier communications from the examiner should be directed to Raymond J. Bayerl whose telephone number is (571) 272-4045. The examiner can normally be reached on M - Th from 9:00 AM to 4:00 PM ET.

8. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca, can be reached on (571) 272-4048. All patent application related correspondence transmitted by FAX **must be directed** to the central FAX number (571) 273-8300.

9. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-2100.

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173
16 August 2005